AMENDMENT TO THE DRAWINGS:

Subject to the Examiner's approval, please substitute FIGs. 1-28 with the attached substitute FIGs. 1-28.

```
        beq r 3, r4, L2
        li r5, 0
        j L 1
L2:     sll r6, r10, 2
        li r5, 1
L1:     move r2, r5
```

Fig. 1 (PRIOR ART)

```
(1)  pseq <p1, p2>, r3, r4
(2)  <p2> li r5, 0
(3)  <p1> sll r6, r10, 2
(4)  <p1> li r5, 1
(5)  move r2, r5
```

Fig. 2 (PRIOR ART)

Fig. 3

```
                    ┌─────────────┐
                    │    START    │
                    └──────┬──────┘
                           │
          ┌────────────────────────────────────┐
          │         FETCH INSTRUCTIONS         │── S11
          └────────────────┬───────────────────┘
                           │
          ┌────────────────────────────────────┐
          │       DECODE THE INSTRUCTIONS      │── S12
          └────────────────┬───────────────────┘
                           │
          ┌────────────────────────────────────┐
          │   OBTAIN A TASK WINDOW NUMBER OF   │── S13
          │         THE INSTRUCTIONS           │
          └────────────────┬───────────────────┘
                           │
          ┌────────────────────────────────────┐
          │  ASSIGN THE INSTRUCTIONS TO        │── S14
          │  CORRESPONDING QUEUE IN THE        │
          │  EXECUTION BUFFER                  │
          └────────────────┬───────────────────┘
                           │
          ┌────────────────────────────────────┐
          │  DECIDE OPERAND OF THE INSTRUCTIONS │── S15
          └────────────────┬───────────────────┘
                           │
          ┌────────────────────────────────────┐
          │       EXECUTE THE INSTRUCTION      │── S16
          └────────────────┬───────────────────┘
                           │
```

COMMIT INSTRUCTION? — S17

YES

NO

STORE INSTRUCTION? — S18

YES

NO

S19 — ACCEPTANCE · REJECTION

S20 — UPDATE THE LOCAL REGISTER

S21 — REGISTER IN THE STORE BUFFER

END
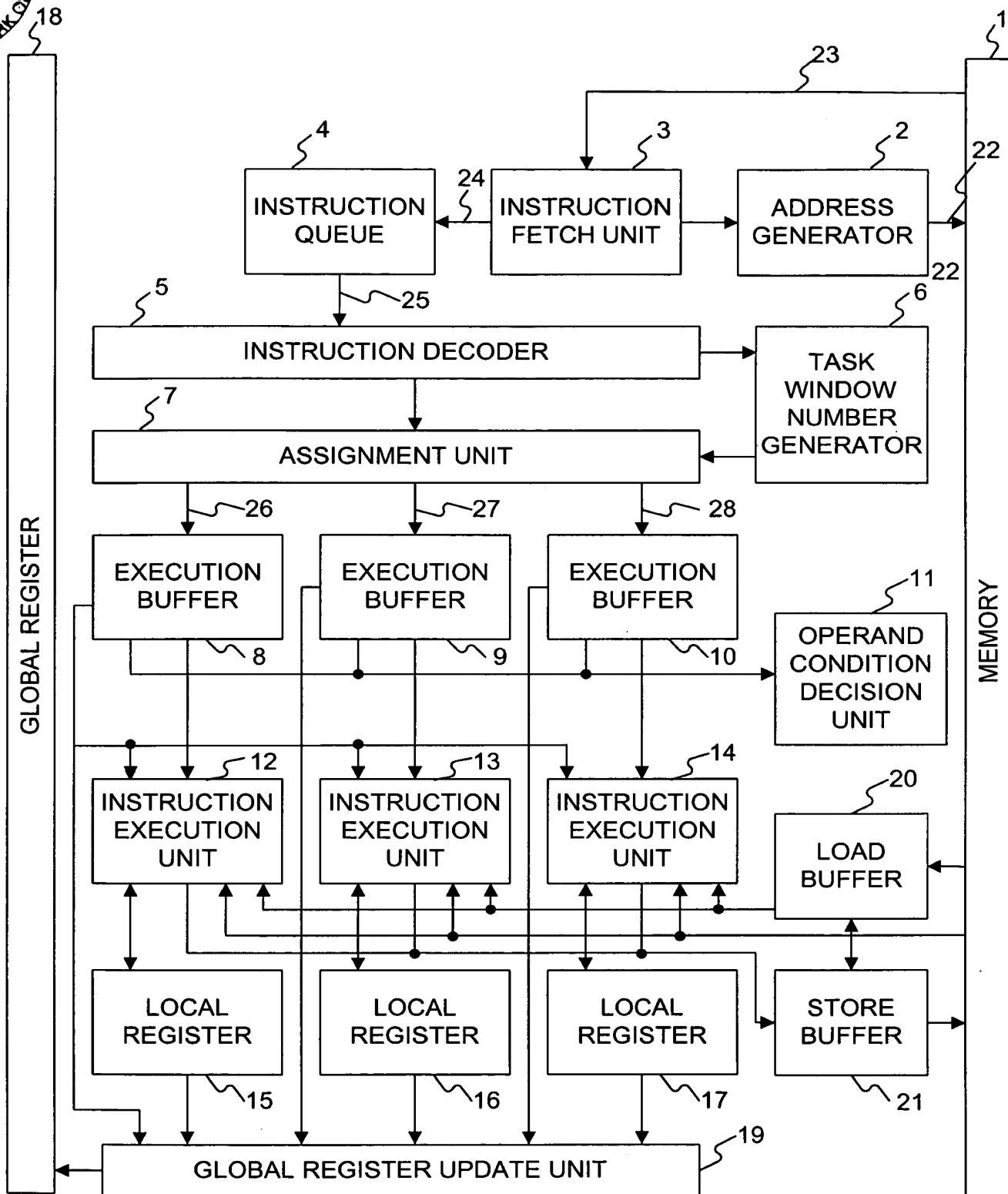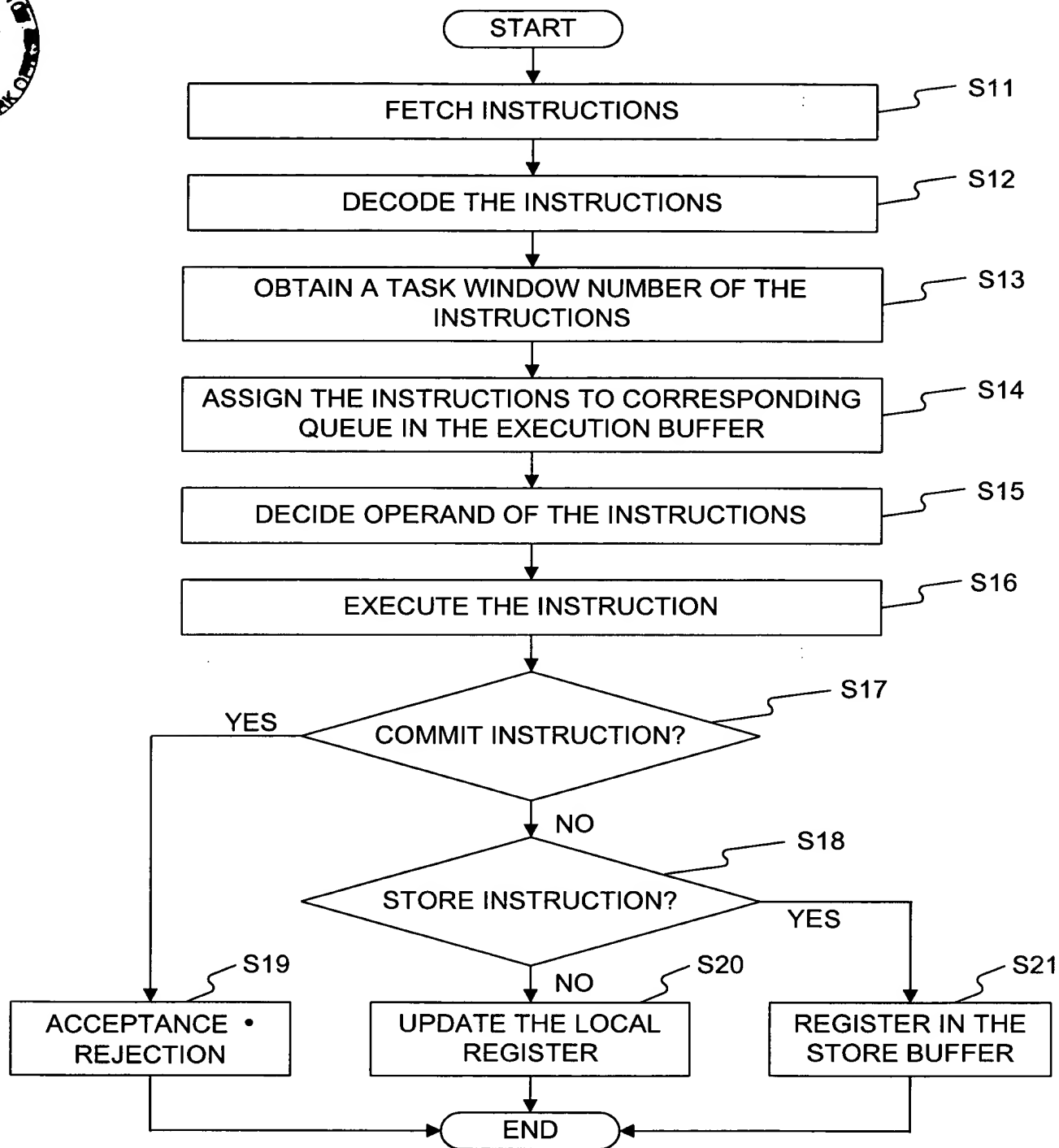
Fig. 4
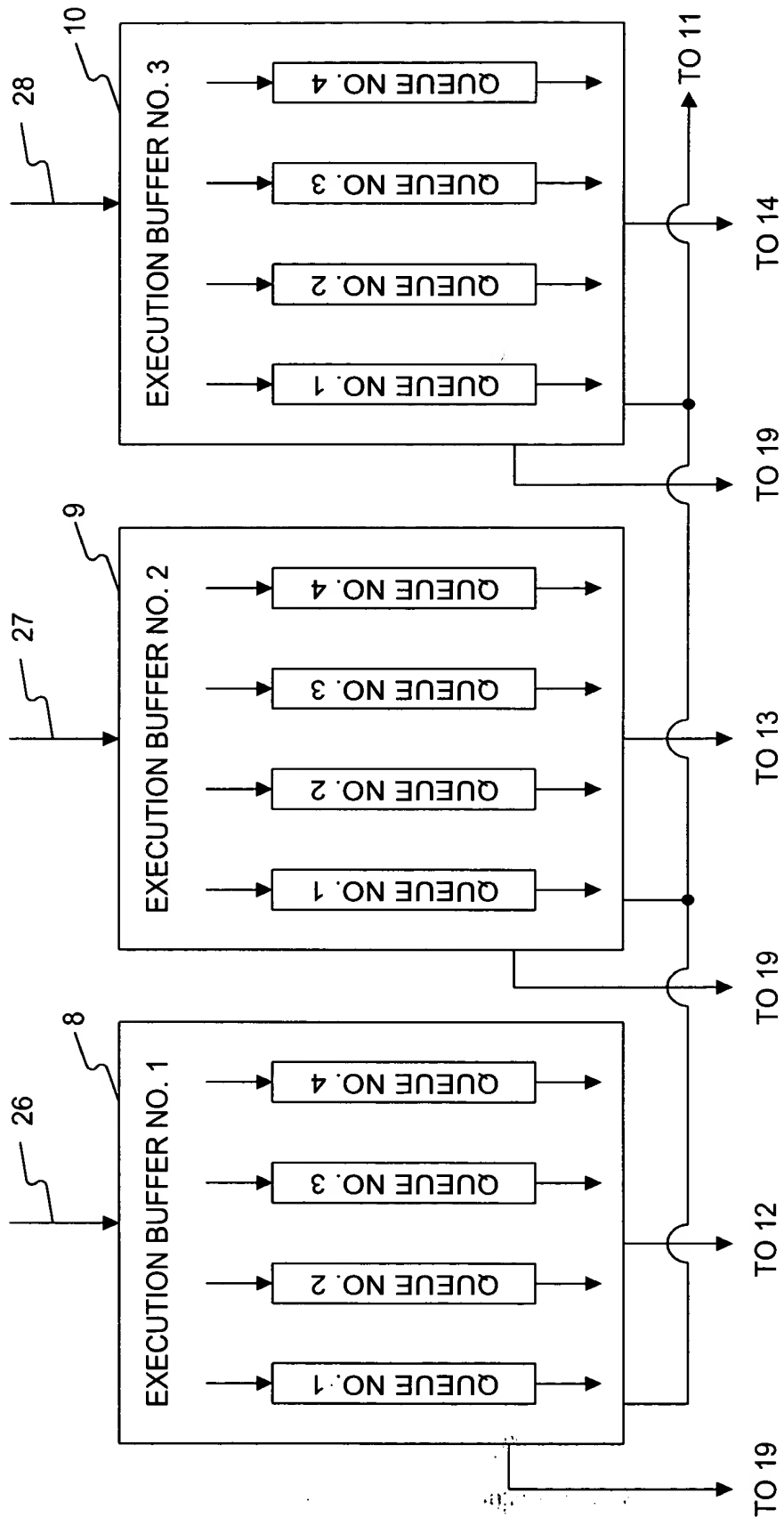
Fig. 5
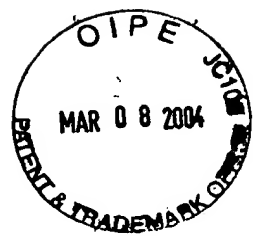
CLASS OF INSTRUCTION	OPERAND DATA	EXCEPTIONAL DATA
l w	7, 3, 0 (0)		← tail
s l l	3, 4, 0 (6)		← execute
a d d i u	4, 5, 0 (100)		← commit
a d d u	5, 6, 0 (0)		← head

Fig. 6

REGISTER	VALUE	VALID
$ 1	100	1
$ 2	15	0

EXAMPLE OF PROGRAM CODE
li   P : $ 2, 2

IN CASE OF DECODING, THE VALID OF REGISTER "$2" IS SET "0" BY PRODUCTION FLAG "P:".

SW $ 2, 0   ($ 4)

THE REGISTER "$2" IS NON-USE UNTIL THE VALID IS CHANGED TO "1".

IF THE VALUE "15" IS CHANGED TO "2", THE VALID "0" IS CHANGED TO "1".
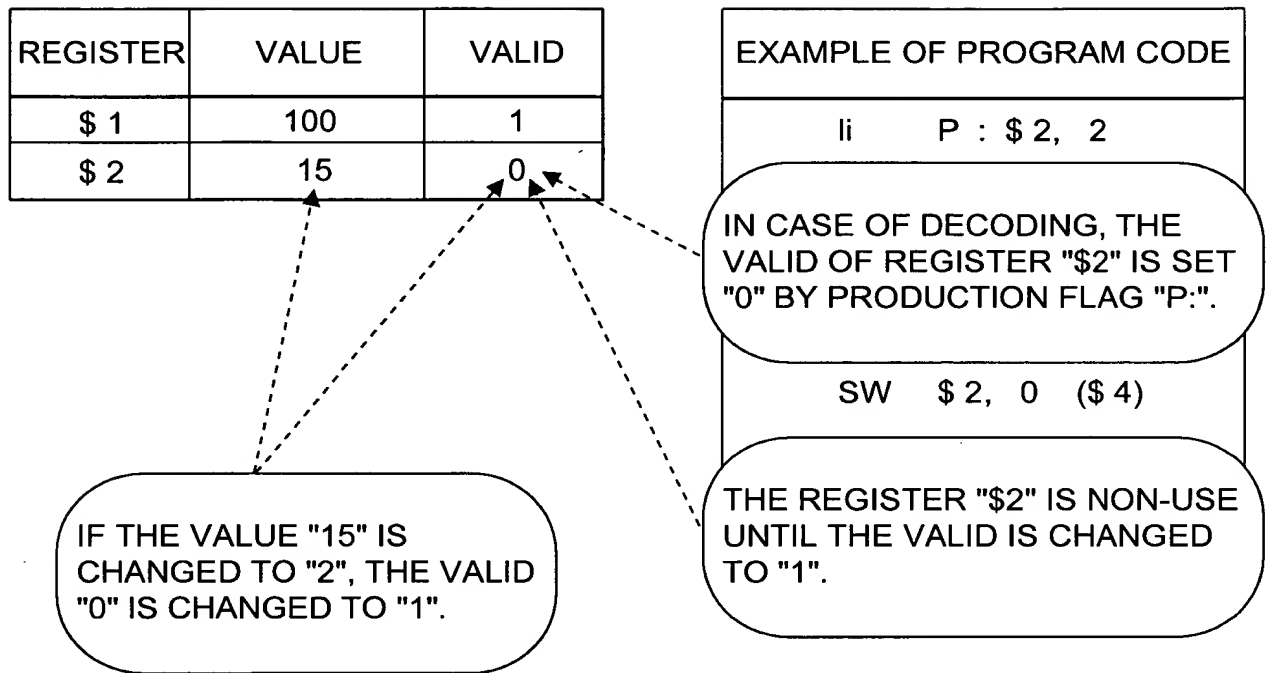
Fig. 7
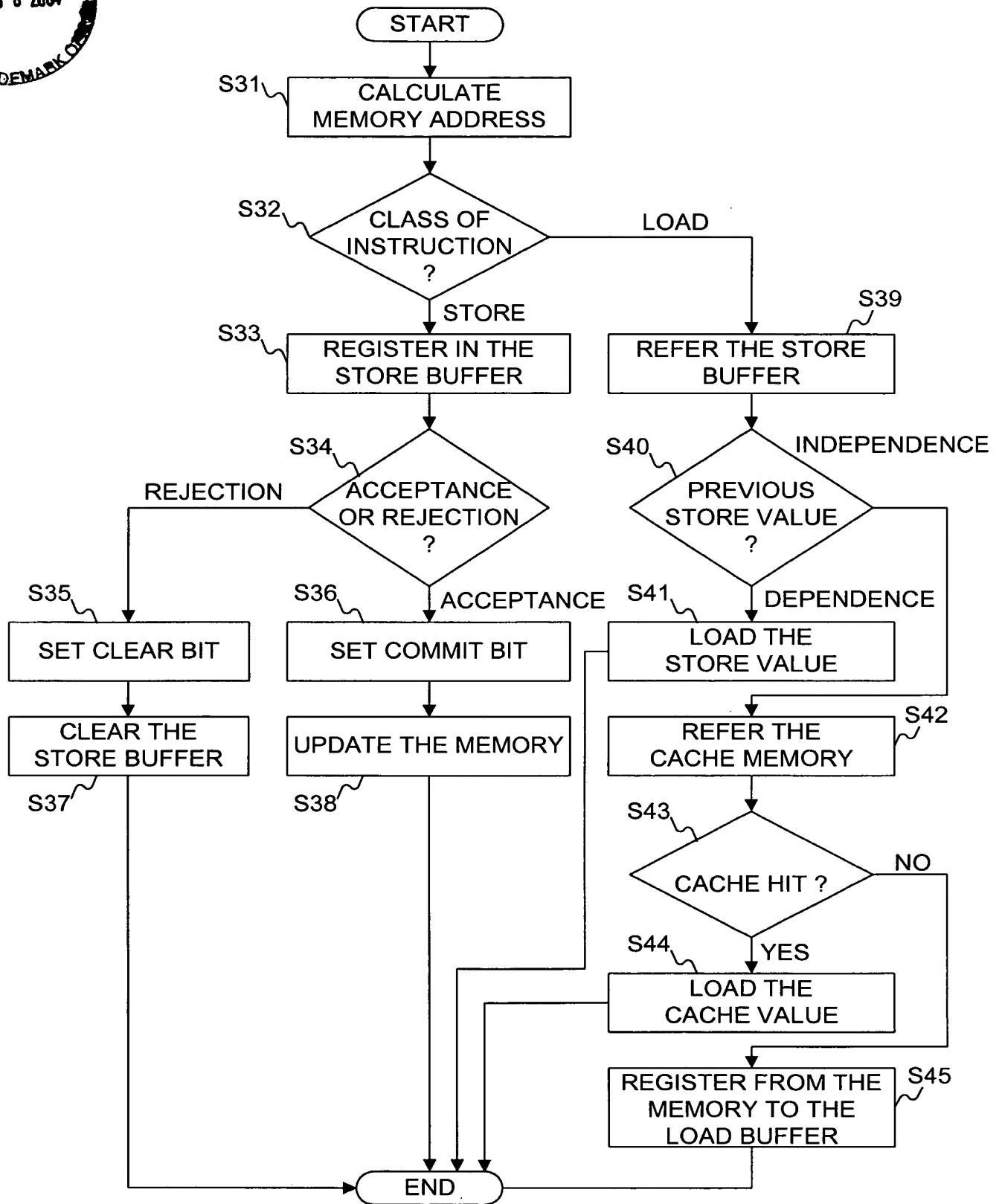
Fig. 8

STORE ADDRESS	STORE VALUE	STORE WIDTH	TWID	CMT	CLR	
10046e3c	3d70	H	1	0	0	← tail
10002a1e	1f	B	0	0	1	
10015d60	1eb94	W	0	1	0	← head

Fig. 9

LOAD ADDRESS	REGISTER FIELD	valid	RFL	
100143e0	02,08,15,22,10,34,01,05	00101001	0	← tail
10003160	03,05,14,13,07,04,05,08	01000101	1	← head

Fig. 10

START

S51 — SEARCH ENTRY OF THE LOAD BUFFER

S52 — CACHE ADDRESS ? — DIFFERENT

EQUAL

S56 — REGISTER FIELD ? — VALID

INVALID

S53 — REFILL BIT ? — "1"

"0"

S54 — MERGE IN CORRESPONDING ENTRY

S55 — CREATE NEW ENTRY

END
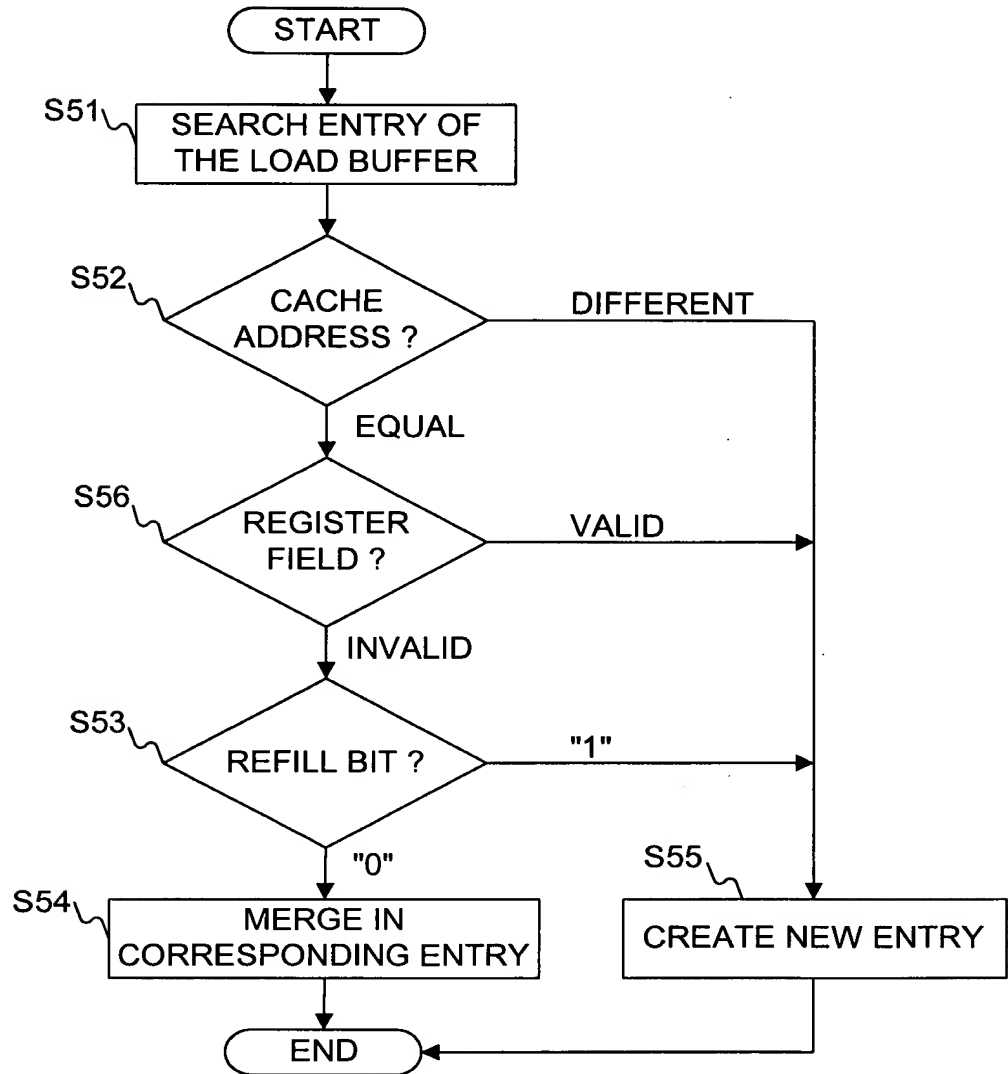
Fig. 11

TASK NUMBER	LOAD ADDRESS	LOAD WIDTH	CNF	VLD
3	1000923a	H	0	1
4	1001024c	W	1	0
2	1002149d	B	1	1

Fig. 12

```
l b u. e      $ 3, C h_1_G l o b ($ 0)
b n e         $ 3, $ 6, $ L48
l w. e        $ 5, I n t_G l o b ($ 0)
a d d i u     $ 7, $ 7, - 1
m o v e       $ 2, $ 0
s u b u       $ 3, $ 7, $ 5
s w           $ 3, 0 ($ 4)

$ L48:
a d d i u     $ 6, $ 0, 5
```
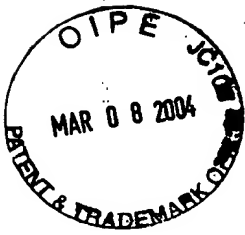
Fig. 13

```
1 : l b u. e      $ 3, Ch_1_G l o b ($ 0)
2 : l w. e        $ 5, I n t_G l o b ($ 0)
2 : a d d i u     $ 7, $ 7, - 1
3 : m o v e       $ 2, $ 0
2 : s u b u       $ 3, $ 7, $ 5
2 : s w           $ 3, 0 ($ 4)
1 : c m t. n e    $ 3, $ 6,
                  |1, 2, 3|, |2, 3|
1 : a d d i u     $ 6, $ 0, 5
```

Fig. 14

```
        l w.  e        $ 4, I n t_G l o b ($ 0)
        s l t i        $ 3, $ 4, 101
        b n e          $ 3, $ 0, $ L14
        j              $ L 9
        s w            $ 0, 0 ($ 56)
$ L 1 4 : a d d i u    $ 3, $ 0, 3
```

Fig. 15

```
1 : l w.  e         $ 4, I n t_G l o b ($ 0)
2 : s w             $ 0, 0 ($ 56)
1 : s l t i         $ 3, $ 4, 101
1 : b c m t. 'n e   $ 3, $ 0, $ L14,
                    |1, 2|, |2|
1 : j               $ L9
$ L 1 4 : 2 : a d d i u    $ 3, $ 0, 3
```

Fig. 16

```
            l w             $ 3, 0 ($ 57)
            s w             $ 16, 16 ($ sp)
$ L41 : l w             $ 9, 0 ($ 11)
            a d d i u       $11, $11, 16
            s w             $ 9, 0 ($ 3)
            b n e           $ 11, $ 10, $ L41
```
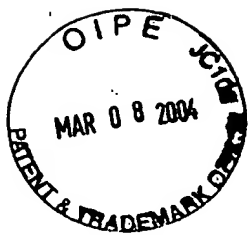
Fig. 17

```
1 : l w             $ 3, 0 ($ 57)
3 : s w             $ 9 : P, 0 ($ 11)
3 : a d d i u       $ 11, $ 11, 16
4 : s w             $ 9, 0 ($ 3)
2 : s w             $ 16, 16 ($ sp)
3 : l c m t. ne     $ 11, $ 10, $ L41,
                    |3, 4|, |1, 2|
```

Fig. 18

```
1 : l w             $ 3, 0 ($ 57)
1 : a d d u         $ 9, $ 3, $ 4
3 : d. a d d u      $ 9, $ 3, $ 4
3 : s w             $ 9, 0 ($ 57)
3 : d c m t         $ 9, |3|
```

Fig. 19

REPLACEMENT SHEET

TASK
WINDOW
NUMBER

TASK
NUMBER

INSTRUCTION
NUMBER

```
          1 : m o v e   P : $_5, $ 4                      [1]
          1 : a d d i u   P : $_1, $ 0, 65                 [2]
          1 : a d d i u   P : $ 5, $ 0, 1                  [3]
          1 : l w   $_1, 0 ($_5)                           [4]
          1 : n o p                                        [5]
          1 : n o p                                        [6]
          1 : a d d i u   P : $ 6, $_1, 10                 [7]
          1 : n o p                                        [8]
    1     1 : n o p                                        [9]
      $ L46:
          3 : a d d i u   $6, $ 6, -1                       [10]
          2 : l b u . e  $_2, C h 1_G l o b ($ 0)          [11]
          3 : l w .  e   P : $ 1_3, I n t_G l o b ($ 0)    [12]
          3 : s u b u   P : $_4, $ 6, $_3                  [13]
          2 : b c m t . n e  $_2, $_1, $L48                [14]
                {1_1, 1_3, 2_1, 2_2, 3_1, 3_3, 3_4}, {1_3, 1_4,
      3_3, 3_4}
          4 : m o v e   P : $ 5, $ 0                       [15]
          4 : s w $_4, 0 ($_5)                             [16]
          2 : n o p                                        [17]
          4 : n o p                                        [18]
    2  $ L48:
          3 : n o p                                        [19]
          2 : n o p                                        [20]
          1 : l c m t . n e  $ 5, $ 0, $ L46               [21]
                {1_3, 1_4, 2_2, 3_1, 3_3, 3_4}, {}
          1 : r c m t   $ 31                               [22]
                {1_1}
    3     1 : n o p                                        [23]
          1 : n o p                                        [24]
```
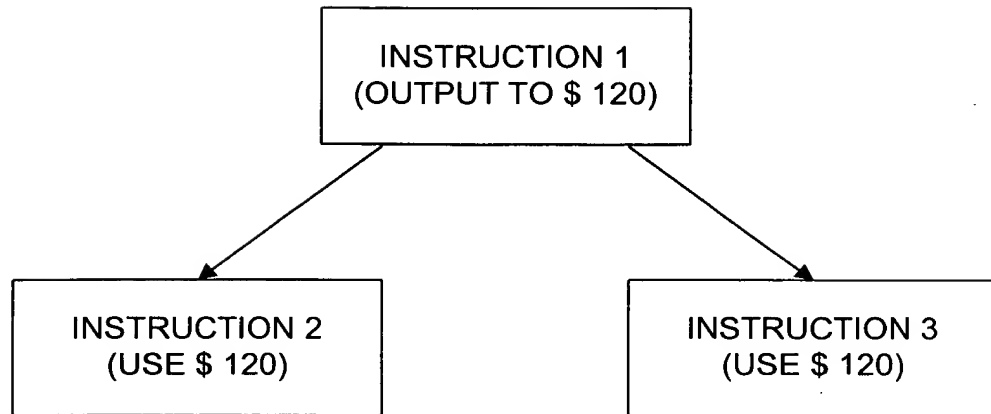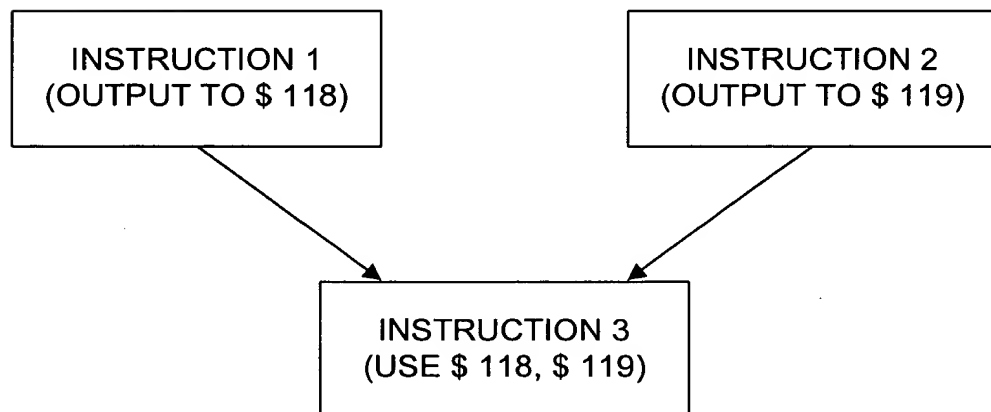
Fig. 20

INSTRUCTION 1
(OUTPUT TO $ 120)

INSTRUCTION 2
(USE $ 120)

INSTRUCTION 3
(USE $ 120)

Fig. 21

INSTRUCTION 1
(OUTPUT TO $ 118)

INSTRUCTION 2
(OUTPUT TO $ 119)

INSTRUCTION 3
(USE $ 118, $ 119)

Fig. 22

```
reflect :
  move    $114, $6 : move $113, $5; move $112, $4
  slti    $115, $114, 2; nop : nop
  bne     $115, $0, $L109; nop : nop
  lw.e    $117, boardsize ($0); addiu $116, $113.1 : nop
  subu    $113, $117, $116; nop : nop

$L109:
  addiu   $118, $0, 1; nop : nop
  beq     $114, $118, $L111; nop : nop
  addiu   $119, $0, 3; nop : nop
  bne     $114, $119, $L110; nop : nop

$L111:
  lw.e    $121, boardsize ($0); addiu $120, $112, 1: nop
  subu    $112, $121, $120; nop : nop

$L110:
  lw.e    $123, boardsize ($0); nop : nop
  mult    $113, $123; nop : nop
  nop;    nop : nop
  nop;    nop : nop
  mflo    $124; nop : nop
  jr      $31; addu $2, $124, $112; nop
```

Fig. 23

```
(BASIC BLOCK 1)
DATA DEPENDENCE SEQUENCE 1 ----->  move  $114, $6
                                   slti  $115, $114, 2
                                   bne   $115, $0, $L109
DATA DEPENDENCE SEQUENCE 2 ---->   move  $113, $5
DATA DEPENDENCE SEQUENCE 3 ---->   move  $112, $4

(BASIC BLOCK 2)
DATA DEPENDENCE SEQUENCE 4 ----->  lw. e  $117, boardsize ($0)
DATA DEPENDENCE SEQUENCE 5 ---->   addiu  $116, $113, 1
                                   subu   $113, $117, $116

(BASIC BLOCK 3)
DATA DEPENDENCE SEQUENCE 6 ----->  addiu  $118, $0, 1
                                   beq    $114, $118, $L111

(BASIC BLOCK 4)
DATA DEPENDENCE SEQUENCE 7 ----->  addiu  $119, $0, 3
                                   bne    $114, $119, $L110

(BASIC BLOCK 5)
DATA DEPENDENCE SEQUENCE 8 ---->   lw. e  $121, boardsize ($0)
DATA DEPENDENCE SEQUENCE 9 ---->   addiu  $120, $112, 1
                                   subu   $112, $121, $120

(BASIC BLOCK 6)
DATA DEPENDENCE SEQUENCE 10 --->   lw. e  $123, boardsize ($0)
                                   mult   $113, $123
                                   nop
                                   nop
                                   mflo   $124
                                   addu   $2, $124, $112

DATA DEPENDENCE SEQUENCE 11 ---->  jr $31
```
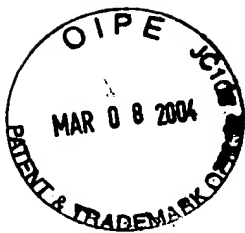
Fig. 24

```
reflect :
  lw.e    $123,boardsize($0);move $114, $6;   move$113, $5
  mult    $113,$123;  slti $115,$114,2;  move $112,$4
  nop;    lw.e $117,boardsize($0);  addiu$116,$113.1
  nop;    bne  $115,$0,$L109;  subu $113,$117,$116

$L109:  mflo  $124;  addiu$118,$0, 1;  aadiu      $119,$0, 3
  nop:    beq  $114,$118,$L111; nop
  nop:    bne  $114,$119,$L110; nop

$L111:  addiu  $120,$112, 1:  lw. e $121,boardsize($0);  nop
  subu   $112,$121,$120;  nop: nop

$L110:
  addiu  $2,$124,$112;  jr $31:  nop
```

Fig. 25

```
reflect :
3 : lw. e $l123, boardsize ($0); 1 : move $l114, $6; 1 : move $l113, $5
3 : mult $l113, $l123; 1 : slti $l115, $l114, 2; 2 : move $l112, $4
3 : lw. e $l117, boardsize ($0); 3 : addiu $l116, $l113, 1
3 : nop; 1 : bne $l115, $0, $L109; 3 : subu $l113, $l117, $l116
3 : nop;

SL109 :
3 : mflo; $l124; 1 : addiu $l118, $0, 1; 3 : addiu $l119, $0, 3
3 : nop; 1 : beq $l114, $l118, $L111; 3 : nop
1 : nop; 1 : bne $l114, $l119, $L110; 1 : nop

SL111 :
1 : addiu $l120, $l112, 1; 1 : lw. e $l121, boardsize ($0); 1 : nop
1 : subu $l112, $l121, $l120; 1 : nop; 1 : nop

SL110 :
1 : addiu $2, $l124, $l112; 1 : jr $31; 1 : nop
```

Fig. 26

```
reflect:
3::lw. e $123, boardsize ($0); 1::move $114,$6; 1::move $113, $5
3::mult $113, $123; 1::slti $115,$114, 2; 2::move $112,$4
3::nop; 3::lw. e $117, boardsize ($0); 3::addiu $116,$113
3::nop; 1::cmt. ne $115, $0,; 3::subu$113,$117,$116
                {2_1, 2_3, 3_1, 3_2, 3_3|, |2_3, 3_3}
3::mflo; 1::addiu $118,$0,1; 3::addiu$119, $0, 3
3::nop; 1::bcmt. eq $114, $118, $L111,; 3::nop
                {1_3, 2_1, 3_3, 3_4|, |3_3, 3_4}
1::nop; 1::bcmt. ne$114, $119,$L110,; 1::nop
                {1_1, 2_1, 3_1|, }{

$ L111:
1::addiu $120,$112, 1; 1::lw. e$121, boardsize ($0); 1::nop
1::subu $112,$121, $120; 1::nop; 1::nop; 1::nop

$ L110:
1::addiu $2,$124, $112; 1::rcmt$3 1,; 1::nop
                {1_1, 2_1}
```

Fig. 27

```
reflect:
 3:lw. e $_1. boardsize ($0); 1:move P:$_1. S6: 1:move P:$3.$5
 3:mult $3, $_1; l:slti $_1. $_1.2: 2:move P:$7, $4
 3:nop: 3:lw. e P:$3_2, boardsize ($0): 3:addiu $_1. $3. 1
 3:nop: 1:cmt. ne $_1, $0,: 3:subu $3, $_2. $_1
          |2_1, 2_3, 3_1, 3_2, 3_3|, |2_3, 3_3|
 3:mflo P:$_2; 1:addiu $_2, $0, 1: 3:addiu P:$2_3, $0, 3
 3:nop: 1:bcmt. eq $_1, $_2, $L111,: 3:nop
          |1_3, 2_1, 3_3, 3_4|, |3_3, 3_4|
 1:nop: 1:bcmt, ne $_1, $_3, $L110,: 1:nop
          |1_1, 2_1, 3_1|, ||
$L111:
 1:addiu $_3. $7. 1: 1:lw. e P:$1_4, boardsize ($0): 1:nop
 1:subu $7, $_4. $_3; 1:nop: 1:nop
$L110:
 1:addiu $2. $_2, $7: 1:rcmt $31,; 1:nop
                        |1_1, 2_1|
```

Fig. 28